AMENDMENT TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Please amend the claims as follows:

1. (Currently Amended) A central processing apparatus for assigning instructions of a program to a plurality of buffers each connected to one of a plurality of execution units, the plurality of execution units each executing the instructions by accessing a memory and a global register,

wherein the program comprises a plurality of instruction sequences, each instruction sequence comprises a plurality of instructions not executable in parallel because of data dependency,

wherein an instruction sequence speculatively executable is located before an instruction sequence not speculatively executable, the plurality of instruction sequences are aligned in correspondence with each of the plurality of buffers, a task number representing the instruction sequence and a corresponding buffer is assigned to each instruction, a condition instruction is replaced by a commit instruction, the commit instruction includes a condition, task ~~number~~ numbers to be accepted if the condition is not satisfied and task numbers to be rejected if the condition is ~~satisifed~~ satisfied,

the central processing apparatus, comprising:

a task window number generator configured to assign a task window number to a plurality of instruction sequences in a task window, the commit instruction being at the end of the task window;

an assignment unit configured to assign each instruction of aligned instruction sequences to each of the plurality of buffers by referring to the task number;

a register update unit configured to update data in a register number accessed by a particular instruction sequence in a task window if the particular instruction sequence is accepted by the commit instruction in the task window; and

a memory update unit configured to update data in a memory address accessed by a particular instruction sequence in a task window if the particular instruction sequence is accepted by the commit instruction in the task window.

2. (Currently Amended) The central processing apparatus according to claim 1,

wherein the instruction sequence of data production or data consumption includes a flag representing possession of the register number in the global register.

3. (Previously Presented) The central processing apparatus according to claim 2,

wherein the plurality of instruction sequences is aligned so that the number of aligned instructions is equal to the number of the plurality of execution units.

4. (Previously Presented) The central processing apparatus according to claim 3,

wherein the instruction sequence speculatively executable is the instruction sequence to be executed if the condition is not satisfied and the instruction sequence to be executed if the condition is satisfied, and

wherein the instruction sequence not speculatively executable is the instruction sequence including the commit instruction.

5. (Original) The central processing apparatus according to claim 4,

wherein the commit instruction represents a branch condition instruction, and additionally includes a branch address of the instruction if the condition is satisfied.

6. (Original) The central processing apparatus according to claim 3,

wherein the commit instruction in the task window represents a loop condition instruction, and includes a loop condition, the task numbers representing a loop, and the task numbers to be accepted in case of the first loop only.

7. (Original) The central processing apparatus according to claim 3,

wherein the global register comprises a plurality of register numbers, and the flag of the instruction of data production or data consumption represents the register number of non-use for other instructions until the instruction is completely executed.

8. (Original) The central processing apparatus according to claim 7,

further comprising an instruction decoder configured to decode a plurality of the instructions in order, and supply each instruction to said assignment unit along with operand data, the task number, and the task window number to which the instruction belongs.

9. (Original) The central processing apparatus according to claim 8, wherein,

when said instruction decoder decodes the instruction including the flag,

said instruction decoder sets the register number in the global register represented by the flag as non-use for the other instructions.

10. (Original) The central processing apparatus according to claim 3,

wherein each of the plurality of buffers comprises a plurality of queues, and each of the plurality of queues exclusively stores the instructions of predetermined task number by first in first out.

11. (Previously Presented) The central processing apparatus according to claim 10,

wherein said assignment unit assigns each instruction of the aligned instruction sequences to the queue in the execution buffer corresponding to the task number of each instruction and

further comprising an operand condition decision unit which repeatedly selects one instruction from each instruction at a head position of each queue in the execution

buffer by priority order, and indicates the execution buffer to transfer the one instruction to the execution unit connected to the execution buffer.

12. (Original) The central processing apparatus according to claim 1,

wherein said task window number generator increments the task window number by one when the commit instruction is detected as the last instruction in the instruction sequence, and assigns an incremented task window number to the instruction sequences consisting of the instruction sequences from an instruction just behind the commit instruction to the next commit instruction in the program.

13. (Original) The central processing apparatus according to claim 9,

further comprising a plurality of local registers respectively connected to each of the plurality of execution units,

wherein each execution unit executes the instruction by accessing a respective local register in order to temporarily preserve the execution result of the instruction.

14. (Previously Presented) The central processing apparatus according to claim 13, wherein,

if a particular instruction sequence is accepted by execution of the commit instruction in the task window,

said register update unit updates data in the register number of the global register represented by the flag in the particular instruction sequence using the execution result preserved in the local register.

15. (Original) The central processing apparatus according to claim 13, wherein,

if a particular instruction sequence is rejected by execution of the commit instruction in the task window,

said register update unit does not update data in the register number of the global register represented by the flag in the particular instruction sequence.

16. (Original) The central processing apparatus according to claim 1,

wherein said memory update unit temporarily preserves the execution result of a store instruction in a particular instruction sequence executed by the execution unit.

17. (Original) The central processing apparatus according to claim 16, wherein,

if the particular instruction sequence is accepted by execution of the commit instruction in the task window,

said memory update unit updates data in the address of the memory represented by the store instruction using the preserved execution result.

18. (Original) The central processing apparatus according to claim 16, wherein,

if the particular instruction sequence is rejected by execution of the commit instruction in the task window,

said memory update unit does not update data in the address of the memory represented by the store instruction.

19. (Original) The central processing apparatus according to claim 16, further comprising a load buffer to temporarily preserve a load instruction in order to read data from the memory or said memory update unit.

20. (Original) The central processing apparatus according to claim 19, wherein,

when the execution unit executes the load instruction in a particular instruction sequence accepted by the commit instruction,

the execution unit decides whether the load instruction depends on the execution result of the store instruction in said memory update unit.

21. (Original) The central processing apparatus according to claim 20, wherein,

if the load instruction depends on the execution result of the store instruction in said memory update unit,

the load buffer loads the execution result of the store instruction from said memory update unit.

22. (Previously Presented) The central processing apparatus according to claim 21, wherein,

if the load instruction does not depend on the execution result of the store instruction in said memory update unit,

the load buffer loads data stored in the address of the memory represented by the load instruction.

23. (Previously Presented) A method for generating a program executed by a central processing apparatus for assigning instructions of the program to a plurality of buffers each connected to one of a plurality of execution units, the plurality of execution units executing the instruction by accessing a memory and a global register, comprising:

dividing the program into a plurality of instruction sequences, each instruction sequence comprising a plurality of instructions not executable in parallel because of data dependency;

moving an instruction sequence speculatively executable forward of an instruction sequence not speculatively executable in the program;

aligning the plurality of instruction sequences in correspondence with each of the plurality of buffers;

assigning a task number representing the instruction sequence and a corresponding buffer to each instruction; and

replacing a condition instruction by a commit instruction, the commit instruction including a condition, task numbers to be accepted if the condition is not satisfied and task numbers to be rejected if the condition is satisfied.

24. (Previously Presented) A computer readable memory containing computer readable instructions in a computer for assigning instructions of a program to a plurality of buffers each connected to one of a plurality of execution units, the plurality of execution units executing the instructions by accessing a memory and a global register, comprising:

an instruction unit to divide the program into a plurality of instruction sequences, each instruction sequence comprising a plurality of instructions not executable in parallel because of data dependency;

an instruction unit to move an instruction sequence speculatively executable forward of an instruction sequence not speculatively executable in the program;

an instruction unit to align the plurality of instruction sequences in correspondence with each of the plurality of buffers;

an instruction unit to assign a task number representing the instruction sequence and a corresponding buffer to each instruction; and

an instruction unit to replace a condition instruction by a commit instruction, the commit instruction including a condition, task numbers to be accepted if the condition is not satisfied and task numbers to be rejected if the condition is satisfied.